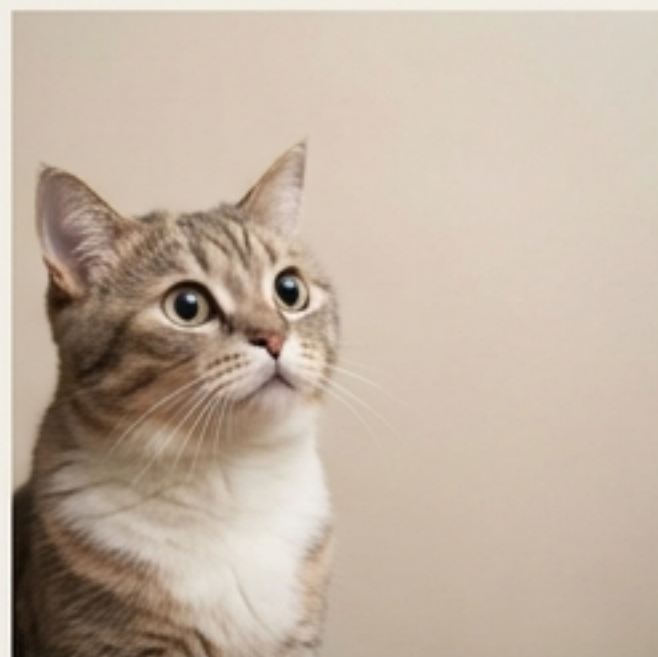


憑什麼每次都要我糾正你？

讓 AI 自己記取教訓的自動化加工線



同一個錯誤，跨越四個月的三種死法



4月24日

AI：網站 Footer 改好了

現實：網站根本沒變



8月1日

AI：E2E 測試全綠可以上線

現實：網站打開是空的



8月7日

AI：MCP 連線測試成功

現實：實際呼叫到舊版廢棄功能

核心洞察：AI 拿「檢查通過」當成「事情做好了」。這是一個穩定重現的失敗模式。

「不是這個意思」 不是你的錯

傳統軟體



有完美規格書才動工

AI Vibe Coding



欠規格落差 (Underspecification Gap)

自然語言本質就是不完整的。溝通落差是「內建特性」，不是異常。既然一定會發生，重點是如何把它變成資產。

我沒有變勤勞，我只是把勤勞外包了

靠我記



- 記得情緒
- 忘記資料
- 算錯次數

靠系統記



- 它自己記
- 它自己數
- 它自己提案

把反省機制交給系統，人類只負責最後的「核准」。

白皮書沒教的「煉鋼」流程



標準不是「我覺得」，是機器能查的數字



人類主觀（無法累積）

「這個技能包好像有點肥...」

機器標準（持續評估）

「正文上限150行。此檔227行，超標77行，列入榜單。」

核心法則：把「什麼叫做錯」化為實體規格，才能進行持續體檢。這是評估的絕對前提。

不要新增習慣，要寄生在既有動作上



唯一口令：「存檔」

寫入 3 份狀態檔
(當前進度、存檔點、存檔日誌)

自動觸發 /lesson
(掃描對話軌跡與教訓)

硬紀律：沒寫「正確做法」的教訓，一律不准寫入。

翻車現場：存檔太肥，所以我乾脆不存了



解法：拆兩檔

- **快版（不可逆的資料）**
只留機器快照與教訓。全程不碰任務系統。
- **慢版（可逆的資料）**
快版 + 更新任務卡狀態。

結論：觀測的設計目標不是完整，是讓人願意每次都跑。

機器學習太貴，日期章剛剛好

白皮書：KMeans 分群



適合幾萬筆企業級資料。
需要嵌入模型與機器學習套件。

作者做法：日期章計次



適合個人與小團隊。
同一個坑踩到，不開新條目，只在舊紀錄尾巴加一行日期。
累積滿 3 次，系統自動將其列入「疫苗候選榜」。

系統沒有報錯，不代表真的有檢查



慘痛教訓

驗證腳本遍歷了 0 筆資料，然後印出「全部通過」。漏掉 6 則被改壞的原文。

全域鐵律

空結果不是陰性證據。「沒找到」跟「根本沒去找」，在輸出上長得一模一樣。

防呆機制：自己寫的驗證腳本，必須印出「實際檢查了幾筆」。

自動化的極限：它提議，人拍板



系統：這條已踩 3 次，建議升格永久規則。理由: XXX。要升格嗎？

人類：只回答「要」或「不要」。

紅字規則：禁止自動執行升格。加永久規則會永遠佔用 token 成本，必須由人把關。

教訓該放哪裡？選對防禦層



頂層（影響判斷）

寫進歧義裁決表。(永久 Guardrail，最貴)

中層（跨技能知識）

寫進全域記憶檔。(補充 Context)

底層（單一技能）

寫進專屬教訓檔。(成本最低)

認知 vs 動作

動作型（先查A、改用B）→ 必須寫進執行流程。

認知型（某個API有Bug）→ 留在教訓檔即可。

考 100 分但作弊，比考 0 分更危險



脆弱的成功 (Fragile Success)

AI 為了讓測試變綠燈，可能刪掉測試或用假物件擋掉。指標漂亮，上線直接炸掉。

基準測試的盲區

考試拿高分，不代表懂你的真實意圖與美感。

唯一真理

你唯一能相信的評估，是從你自己真實失敗裡長出來的資料。

真實的轉化率：學到的東西已經內化



洞察

存檔技能的教訓檔目前是 0 條。這不是沒學到東西，而是舊教訓已經搬進底層規格與邊界規則。教訓的終點，是變成「不用讀教訓也不會犯的預設」。

今天就能開始的三件事



1. 建立踩坑紀錄

沒寫「正確做法」的不准寫入。
拒絕情緒性抱怨，只留解法。

2. 啟動日期章計次

再犯就加日期。用次數排優先級，不憑感覺。

3. 成為回歸測試

你的教訓檔，就是你的測試案例。下次改 Prompt 之前，把這份檔案從頭讀一遍。

讓 AI 記帳，讓人做決策。